AMENDMENTS TO THE CLAIMS

1.   (Currently Amended) A method for storing and manipulating pointers in a programming language, wherein the programming language supports a native pointer type and standard pointer operations upon a native pointer object of said native pointer type, said method comprising the step of:

     defining a safe pointer type supporting said standard pointer operations; ~~and~~

     performing automatic pointer checking in association with said safe pointer type <u>by checking for a null pointer; and</u>

      <u>performing error processing by performing at least one of:</u>

        <u>generating a warning message without terminating program execution; and</u>

        <u>invoking a user defined error processing routine</u>.

2.   (Cancelled)

3.   (Cancelled)

4.   (Cancelled)

5.   (Cancelled)

6.   (Original) The method of claim 1 further comprising the steps of:
calling a function which returns as its value an object of said safe pointer type; and
performing said standard pointer operations upon said object.

7.   (Original) The method of claim 1 further comprising the steps of:
reading a global variable of said safe pointer type; and
performing said standard pointer operation upon said object.

8.   (Currently Amended) The method of claim 1 further comprising the steps of:
calling a function which returns as its value an object of said safe pointer type, said object encapsulating an improper native pointer object <u>comprising a null pointer</u>; and
performing error processing in response to said improper native pointer object.

9.   (Cancelled)

10.   (Currently Amended) The method of claim 1 further comprising the steps of:

reading a global variable encapsulating an improper native pointer object comprising a null pointer; and

performing error processing in response to said improper native pointer object.

11.   (Currently Amended) A safe pointer class embodied in a computer readable medium and supporting both the storage of and manipulation of a pointer in a programming language, the programming language supporting a native pointer type and standard pointer operations upon a native pointer object of said native pointer type, said safe pointer class comprising:

a safe pointer type configured to support said standard pointer operations; and

said safe pointer type further configured to support automatic pointer checking by checking for a null pointer; and

the safe pointer type further configured to perform error processing by performing at least one of:

generating a warning message without terminating program execution; and

invoking a user defined error processing routine.

12.   (Cancelled)

13.   (Cancelled)

14.   (Cancelled)

15.   (Cancelled)

16.   (Cancelled)

17.   (Currently Amended) A computer program product recorded on computer readable medium for reducing a likelihood of misuse of pointers upon which at least one software application is running, said product comprising:

computer readable means for defining a safe pointer type;

said computer readable means supporting standard pointer operations; and

said computer readable means performing automatic pointer checking by checking for a null pointer; and

said computer readable means performing error processing by performing at least one of:

generating a warning message without terminating program execution; and

invoking a user defined error processing routine.

18.   (Cancelled)

19.   (Cancelled)

20.   (Cancelled)

21.   (New) A method for storing and manipulating pointers in a programming language, wherein the programming language supports a native pointer type and standard pointer operations upon a native pointer object of said native pointer type, said method comprising:

defining a safe pointer type supporting said standard pointer operations; and

performing automatic pointer checking in association with said safe pointer type by checking for improper pointer alignment.

22.   (New) The method of claim 21 further comprising a step of performing error processing.

23.   (New) The method of claim 22 wherein said error processing includes at least one of generating an error message and terminating program execution; generating a warning message without terminating program execution; and invoking a user defined error processing routine.

24.   (New) The method of claim 21 further comprising:

calling a function which returns as its value an object of said safe pointer type, said object operable to encapsulate a misaligned pointer as an improper native pointer object;

performing said standard point operations upon said object; and

performing error processing in response to said improper native pointer object.

25.   (New) The method of claim 21 further comprising:

reading a global variable of said safe pointer type, said global variable operable to encapsulate a misaligned pointer as an improper native pointer object;

performing said standard pointer operation upon said object; and

performing error processing in response to said improper native pointer object.

26.   (New) A computer program product recorded on a computer readable medium for reducing misuse of pointers upon which at least one software application is running, said product comprising:

computer readable means for defining a safe pointer type;

said computer readable means supporting standard pointer operations; and

said computer readable means performing automatic pointer checking in association with safe pointer type by checking for improper pointer alignment.

27.   (New) The program product as claimed in claim 26 wherein said computer readable means is further operable to perform error processing.

28.   (New) The program product as claimed in claim 27 wherein said error processing includes at least one of generating an error message and terminating program execution; generating a warning message without terminating program execution; and invoking a user defined error processing routine.

29.   (New) The program product as claimed in claim 26 wherein said computer readable means is further operable to:

call a function which returns as its value an object of said safe pointer type, said object operable to encapsulate a misaligned pointer as an improper native pointer object;

perform said standard point operations upon said object; and

perform error processing in response to said improper native pointer object.

25.   (New) The program product as claimed in claim 26 wherein said computer readable means is further operable to:

read a global variable of said safe pointer type, said global variable operable to encapsulate a misaligned pointer as an improper native pointer object;

perform said standard pointer operation upon said object; and

perform error processing in response to said improper native pointer object.